

**2CS702 BIG DATA ANALYTICS**

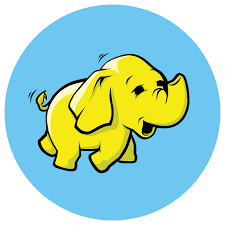
**INNOVATIVE ASSIGNMENT**

***Document Ranking using Apache Hadoop***

**Submitted By**

21BCE009 Aditya Patel

21BCE019 Ayushi Shah



**Table of Contents**

[⚡Motivation](#_heading=h.lw8x8wiw9eya)

[🎯Problem Statement](#_heading=h.hu87eja1caa2)

[📈Use Cases](#_heading=h.tzgw162y40y9)

[💻Experimental setup](#_heading=h.pci0xlx4pod9)

[▶️Work](#_heading=h.m89juqs7o68q) Flow

🏆Output

[📌Challenges and Issues](#_heading=h.ln2wcljgvwo1)

[🥇Conclusion](#_heading=h.6h78gs6c1uye)

[🔗References](#_heading=h.1rg030gdaqlp)

# ⚡Motivation

# The motivation for this project comes from the recognition of the importance of improving the retrieval effectiveness of web search engines within large document libraries. Popular Boolean model of search decreases flexibility in terms of understanding the kind of relationship between the keywords and the contents of documents resulting in ranking the results poorly. As data and document repositories expand at an exponential and unfaltering rate, it is imperative to develop more refined, context sensitive ranking models to increase user entertained and enhance information retrievability. To mitigate these challenges, this project applies the sentence embedding models that provide a higher level of semantic similarity between the query and documents by converting both of them into high dimensional vectors. This approach improves the likelihood of the search engine ability to rank documents on the basis of conceptual similarity not the physical match of keywords. Finally, the motivation behind this project is in shifting the paradigm of search methods, to present a solution that would be as realistic, versatile and applicable across the fields of large scale document relevance.

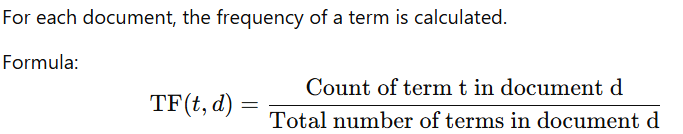
# 🎯Problem Statement

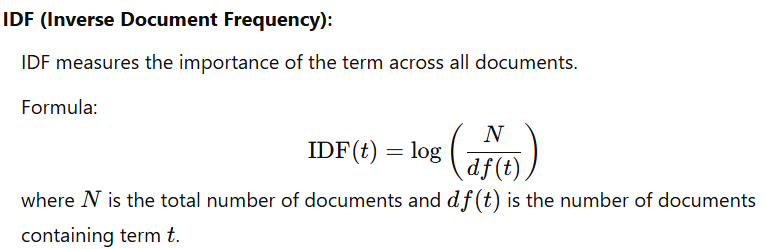
# The major concern this project seeks to solve is the weakness of conventional keyword-based search engines in providing sufficiently relevant rankings of large volumes of documents. The conventional search systems mainly uses a simple keyword matching technique where relations and context between the terms are not well understood. This leads to results that may contain unrelated documents, or other documents which the user would still find to be conceptually related, which in essence defeats the purpose and gives the user a bad experience.

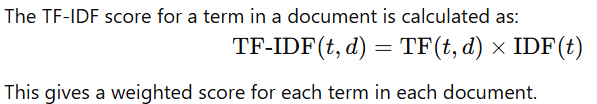
# Hence, search engine must employ a more sophisticated and contextualized method to be able to handle an increased size and complexity of repositories in order to regulate the access to pertinent data. Current approaches are lack the need to accommodate the semantic matching necessary for query results to fully correspond to user intent in different contexts and search scenarios.

# To address this problem, the proposed project aims at creating a document ranking model through the use of sentence embedding with the objective of having both queries and documents represented in high dimensional space. This model will increase the accuracy of ranking to place the most similar documents on top thus enhancing the efficiency of the existing keyword-based search and enhancing the retrieval of information in large databases that contain diverse kinds of data.

# 💡Methodology







# 💻Experimental setup

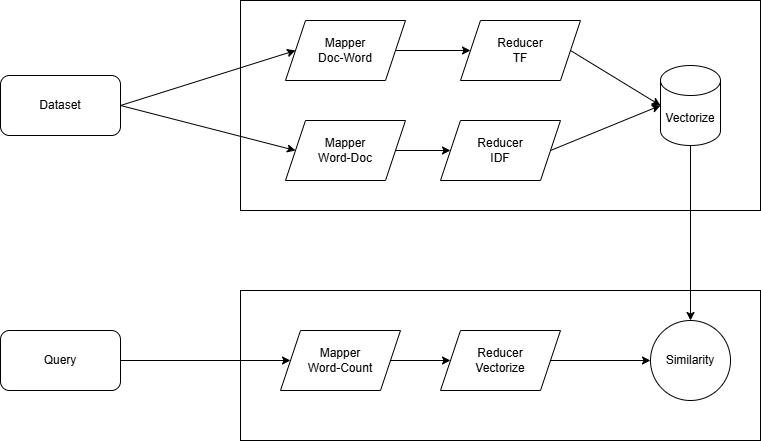
As discussed in the problem statement, the data will be of large volume since the number of documents increase with the high rise of information. Such large volumes of data cannot be stored in the traditional databases.

We have used Apache Hadoop, a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models.

We have used Apache Hadoop in Docker for which Docker was installed.

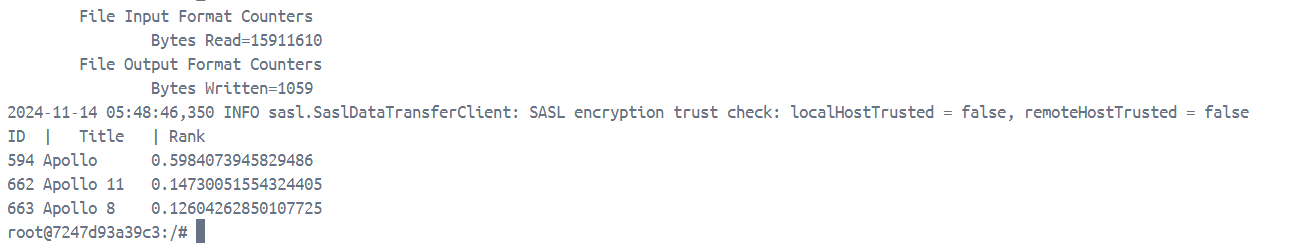
# ▶️Work Flow

* Text files (in .txt format) are loaded into HDFS.
* The code reads the .txt files from HDFS.
* Mapper tokenizes the text and creates (word,1) for each word.
* Reducer calculates the total occurrences i.e term frequency (TF) for each word.
* Then again Mapper fetches the IDF values and computes the TF-IDF for each document.
* The resulting vectors are stored back in HDFS for further use.
* The document vectors stored in HDFS.
* Builds an inverted index by mapping terms (words) to their corresponding documents and TF-IDF scores.
* The user enters a query (as a string).
* The query is tokenized and calculates the TF-IDF vector for the query using the same method as for documents.
* Compares the query vector with document vectors, using cosine similarity to calculate relevance scores for each document.
* Sorts documents based on their relevance to the query.
* After sorting, the code retrieves the top N most relevant documents from HDFS and outputs the document ID, title, and rank.



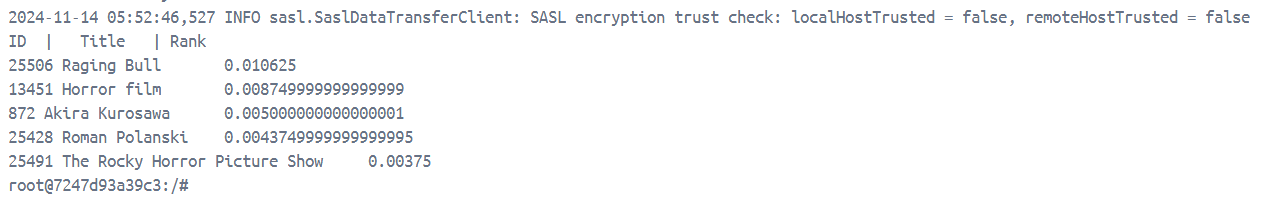
# 🏆Output

Output for fetching top 3 documents for the query “Apollo Space”

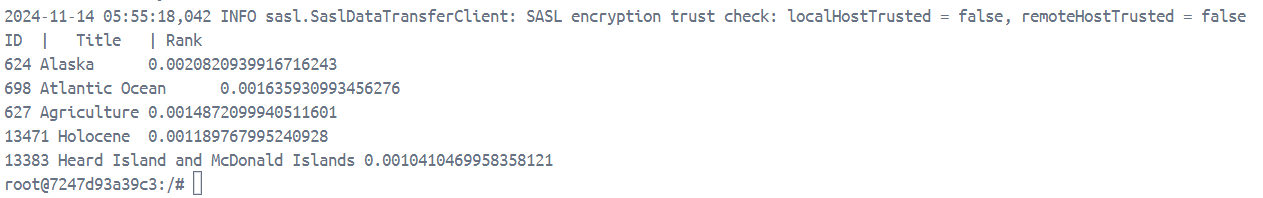
****

****

Output for fetching top 5 documents for the query “Movies”

****

Output for fetching top 5 documents for the query “climate”

****

# 📌Challenges and Issues

Achieving the goals outlined in the problem statement involves addressing several complex challenges and issues. Here are some of the key challenges and issues that need to be overcome:

* **Data Volume and Variety:** A Search engine generates a vast and diverse range of data, including structured and unstructured data from various sources such as live feeds, historical records, social media, and more. Managing and integrating such a diverse dataset can be challenging.
* **Scalability:** The ability to handle the ever-increasing volume of data is critical. As the need and timely presence of certain documents grows, so does the data generated, necessitating scalable infrastructure and systems.
* **Real-time Processing:** For any dynamic and constantly evolving content, real-time data processing is essential. Delayed or outdated information can lead to a suboptimal fan experience.
* **Visualization and Communication:** Presenting the insights derived from big data analysis in a comprehensible and actionable format is vital for their utility by stakeholders.

# 🥇Conclusion

# Altogether, this project utilizes trained sentence embeddings to overcome the drawbacks of search engines while providing a novel, efficient and context-sensitive document ranking system. Through semantic analysis of queries, the proposed model increases the quality of the search query, thereby increasing satisfaction within a range of applications. The capability of a universal sentence encoder fosters the scalability of the system in that it can be applied in large scale data hence it is flexible to be used in various areas of application including; enterprise search and navigation, e-learning, healthcare and customer relationship management. Finally this project provides a basis for future development of information retrieval enhancing the search technology to be smarter, oriented to users more and more.

# 🔗References

|  |
| --- |
| <https://hadoop.apache.org/> |
| <https://hadoop.apache.org/docs/stable/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.html> |
| <https://www.geeksforgeeks.org/mapreduce-combiners/> |
| <https://www.databricks.com/glossary/hadoop-cluster> |